SP24 OCR Capstone Project

1. INTRODUCTION

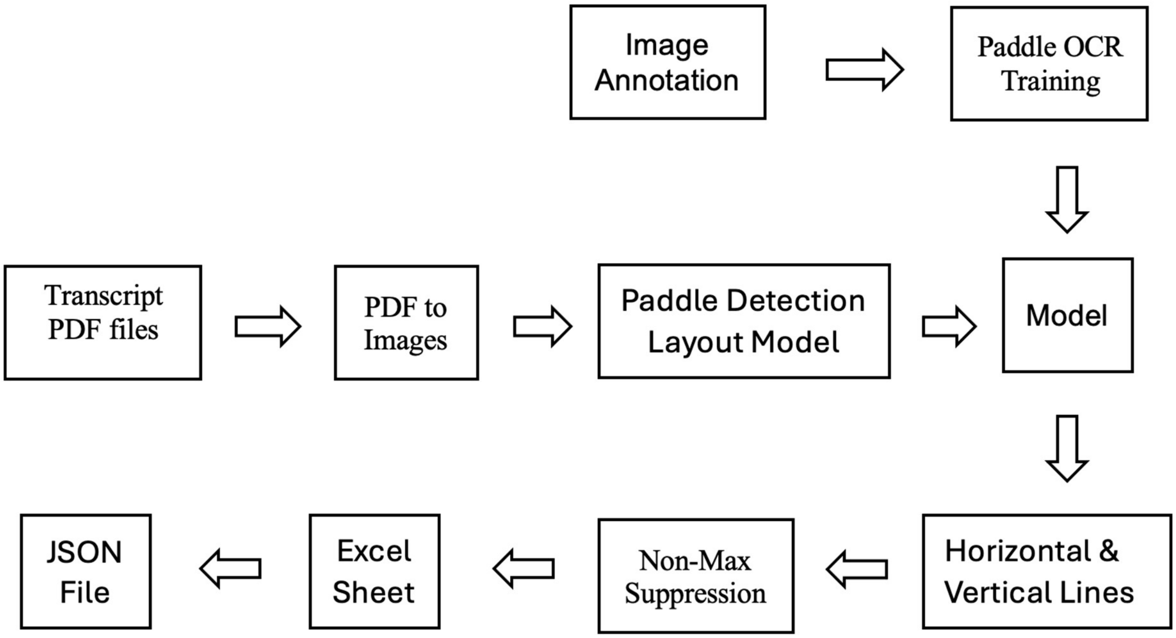
The evaluation and analysis of student transcripts play a critical role in the day-to-day operations of a university. These documents hold a wealth of information essential for informed decision-making in areas such as admissions, scholarship awards, and academic progress monitoring. However, traditionally, this data resides within transcripts themselves, often presented in physical form or scanned images. Extracting this information manually can be a laborious and time-consuming process, hindering efficiency and potentially introducing human error.

This project aims to address this challenge by developing a novel system capable of automatically extracting key data points from university-issued transcripts and converting them into a structured JSON format. This report details the design and implementation of this data extraction system.

This report details the design and implementation of a data extraction system for university transcripts. The system leverages a combination of Layout Parsing and Optical Character Recognition (OCR) technologies. The Layout Parser identifies the structure and organization of the transcript document, while the OCR model recognizes the text within the identified regions. This combined approach allows the system to accurately extract key information from transcripts, such as student names, course details, grades, and awards.

By converting this data into a JSON format, the system facilitates easier access and analysis for university officials and professors. This not only streamlines the decision-making process for admissions and scholarships but also enables the creation of data-driven reports and visualizations to gain deeper insights into student performance and program effectiveness.

2. METHODOLOGIES AND ACTIVATES



Block Diagram

1. Image Annotation:

- Used Label Studio to annotate images and generate a JSON file for training purposes.

2. Paddle OCR Training:

- We fine-tuned the PaddleOCR model on the Transcript dataset to enhance its performance.
- This output model weights is then utilized into further extraction and post processing of the data.

3. PDF TO Images:

- Utilized pdf2image to convert PDF documents into images.

4. Paddle Detection Layout Model:

- Using the PaddleDetectionLayoutModel, we detect various elements within an image, including text, titles, lists, tables, and figures, and extract their corresponding bounding boxes along with the text content.

- From this layout we cropped our images and pass it to OCR model

5. Paddle OCR Model Inference:

- With the Paddle OCR model, text detection and recognition are performed, providing information such as coordinates, text content, and threshold scores.

6. Horizontal & Vertical Lines:

- The OCR model extracts coordinates, which are then used to generate horizontal and vertical lines, ensuring the correct format for tabular data.

7. Non-Max Suppression:

- It determines the necessary selection of rows and columns while ensuring that they do not overlap, storing the resulting data in an array.

8. Excel Sheet:

- The output array is converted into CSV format and saved into an Excel sheet, with handling for duplicate rows and columns.

9. JSON File:

- In the final step, we convert the Excel data into JSON format for each image. This process is repeated for all images in the pdf file.

ALGORITHMS AND METHODS

Image Annotations for Model Training:

Text Extraction from Scanned Documents

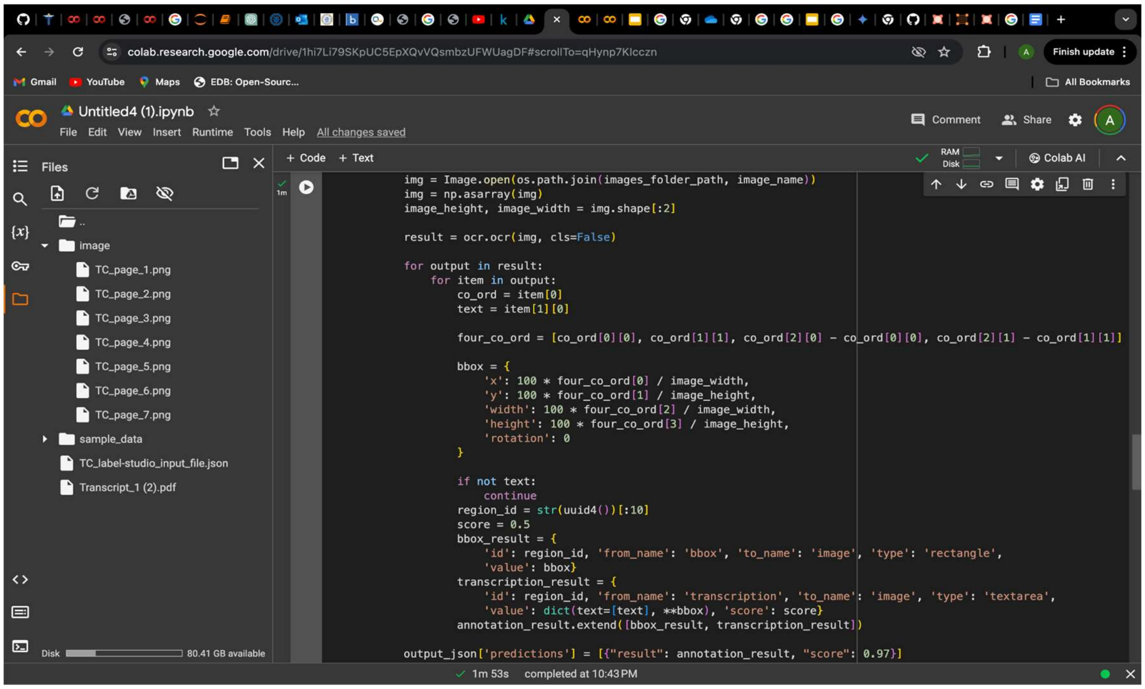
Scanned documents, typically in PDF format and containing transcripts, underwent text extraction using the PaddleOCR framework:

- Utilization of PaddleOCR: The PaddleOCR library, a robust Optical Character Recognition (OCR) tool, was employed to extract text from images of each page within the scanned documents.
- OCR Processing: PaddleOCR processed each page image, detecting and recognizing text to generate machine-readable content.

Image Pre-processing and Annotation Generation

Pre-processing techniques were applied to enhance the quality of the extracted text and facilitate the creation of initial annotations:

- Image Enhancement: Various pre-processing techniques, including resizing, contrast adjustment, and noise reduction, were implemented to optimize image quality.
- Annotation Generation: Bounding boxes were automatically generated around detected text regions by the PaddleOCR model. Corresponding text labels were extracted to create initial annotations.
- JSON File Creation: Annotations, accompanied by the URLs of the processed images, were organized and stored in a JSON file format. This file served as the basis for further annotation refinement.

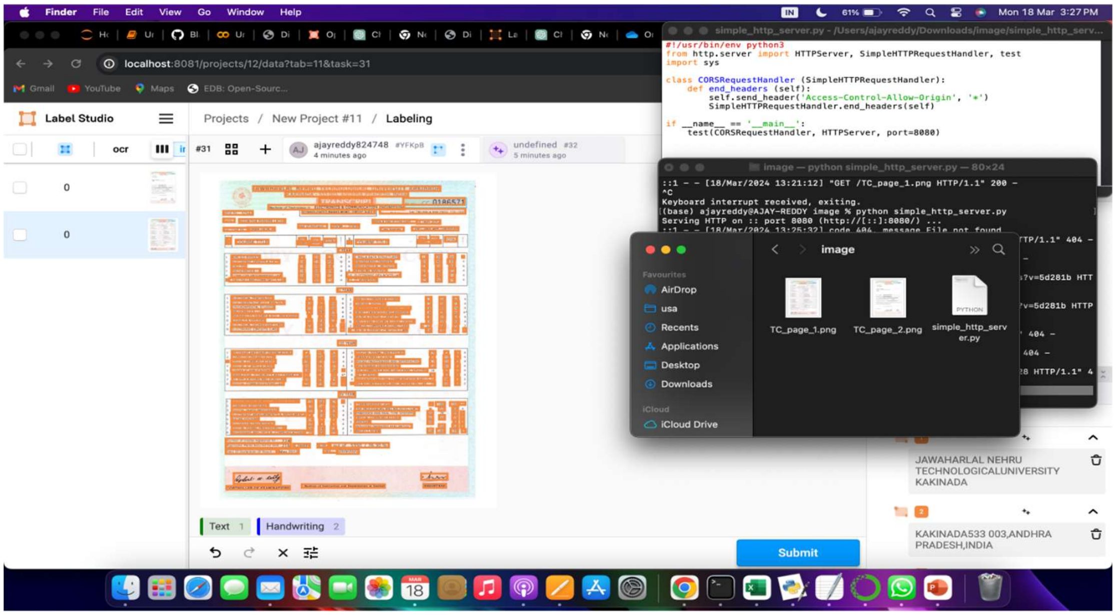


```
        img = Image.open(os.path.join(images_folder_path, image_name))
        img = np.asarray(img)
        image_height, image_width = img.shape[:2]

        result = ocr.ocr(img, cls=False)

        for output in result:
            for item in output:
                co_ord = item[0]
                text = item[1][0]

                four_co_ord = [co_ord[0][0], co_ord[1][1], co_ord[2][0] - co_ord[0][0], co_ord[2][1] - co_ord[1][1]]

                bbox = {
                    'x': 100 * four_co_ord[0] / image_width,
                    'y': 100 * four_co_ord[1] / image_height,
                    'width': 100 * four_co_ord[2] / image_width,
                    'height': 100 * four_co_ord[3] / image_height,
                    'rotation': 0
                }

                if not text:
                    continue
                region_id = str(uuid4())[:10]
                score = 0.5
                bbox_result = {
                    'id': region_id, 'from_name': 'bbox', 'to_name': 'image', 'type': 'rectangle',
                    'value': bbox}
                transcription_result = {
                    'id': region_id, 'from_name': 'transcription', 'to_name': 'image', 'type': 'textarea',
                    'value': dict(text=[text], **bbox), 'score': score}
                annotation_result.extend([bbox_result, transcription_result])

        output_json['predictions'] = [{"result": annotation_result, "score": 0.97}]
```

Annotation Setup

The initial annotations generated by the PaddleOCR model were integrated into the Label Studio platform for annotation refinement:

- JSON File Upload: The JSON file containing initial annotations was uploaded to Label Studio, providing a foundation for subsequent manual annotation tasks.
- Image Loading: Images, along with their associated annotations, were loaded into the Label Studio interface for review and manual editing by project team members.



```python
#!/usr/bin/env python3
from http.server import HTTPServer, SimpleHTTPRequestHandler, test
import sys

class CORSRequestHandler (SimpleHTTPRequestHandler):
    def end_headers (self):
        self.send_header('Access-Control-Allow-Origin', '*')
        SimpleHTTPRequestHandler.end_headers(self)

if __name__ == '__main__':
   test(CORSRequestHandler, HTTPServer, port=8080)
```

```
::1 - - [18/Mar/2024 13:21:12] "GET /TC_page_1.png HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
(base) ajayreddy@AJAY-REDDY image % python simple_http_server.py
Serving HTTP on :: port 8080 (http://[::]:8080/) ...
::1 - - [18/Mar/2024 13:25:32] code 404, message File not found
                                                             TTP/1.1" 404 -
                                                              -
                                                             s?v=5d281b HTT
                                                             ?v=5d281b HTTP
                                                             " 404 -
                                                              404 -
                                                             28 HTTP/1.1" 4
```

JAWAHARLAL NEHRU TECHNOLOGICALUNIVERSITY KAKINADA

KAKINADA533 003,ANDHRA PRADESH,INDIA

Review and Correction

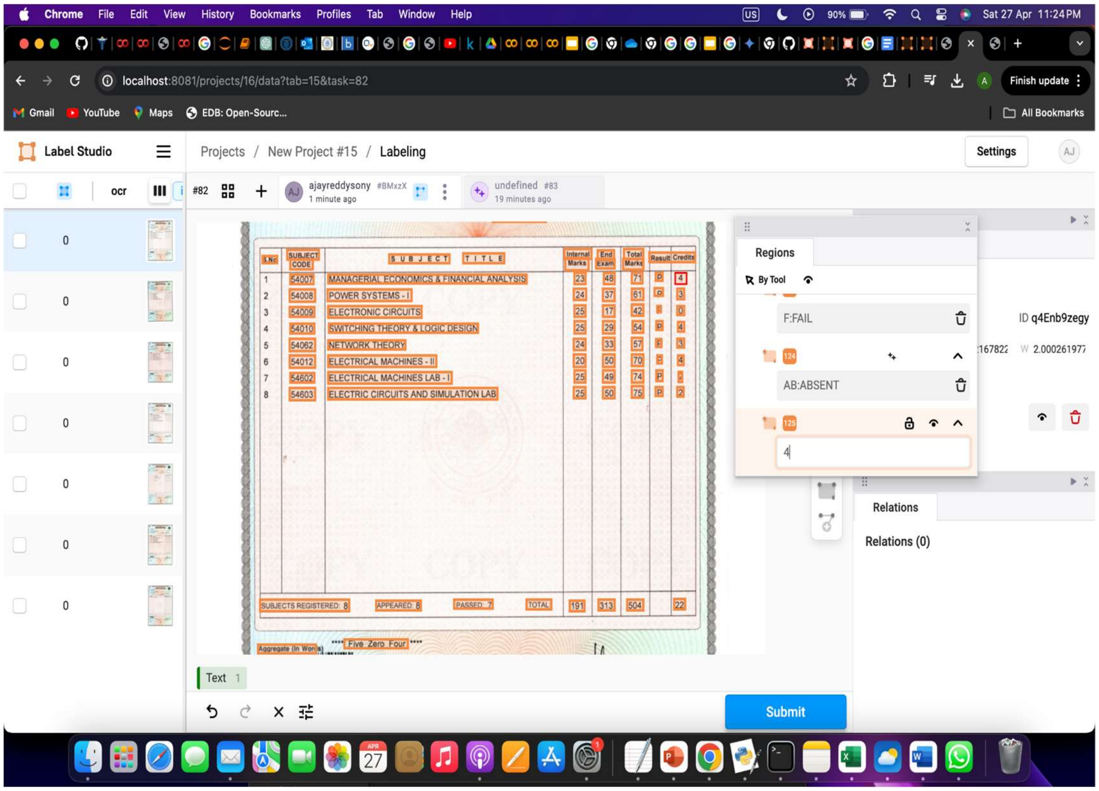
Project team members conducted a thorough review of the initial annotations to identify any inaccuracies or omissions:

- Visual Inspection: Images were visually inspected to pinpoint undetected or inaccurately detected text regions.
- Correction Identification: Undetected words, inaccurately detected text, or other discrepancies were identified for manual correction and refinement.

Manual Annotation

Manual annotation efforts were undertaken to rectify any inaccuracies or omissions identified during the review process:

- Manual Corrections: Team members manually annotated undetected words or corrected inaccurately detected text by drawing bounding boxes or polygons around the relevant regions.
- Text Entry: For each annotation, corresponding text was entered manually to ensure accurate labeling and alignment with the image content.


Label Studio
Projects / New Project #15 / Labeling
Settings
ajayreddysony #BMxzX 1 minute ago
undefined #83 19 minutes ago
Regions
By Tool
Relations
Relations (0)
Text 1
Submit

localhost:8081/projects/16/data?tab=15&task=82

S.No	SUBJECT CODE	SUBJECT TITLE	Internal Marks	End Exam	Total Marks	Result	Credits
1	54007	MANAGERIAL ECONOMICS & FINANCIAL ANALYSIS	23	48	71	P	4
2	54008	POWER SYSTEMS - I	24	37	61	P	3
3	54009	ELECTRONIC CIRCUITS	25	17	42	F	0
4	54010	SWITCHING THEORY & LOGIC DESIGN	25	29	54	P	4
5	54062	NETWORK THEORY	24	33	57	P	3
6	54012	ELECTRICAL MACHINES - II	20	50	70	P	4
7	54602	ELECTRICAL MACHINES LAB - I	25	49	74	P	2
8	54603	ELECTRIC CIRCUITS AND SIMULATION LAB	25	50	75	P	2
SUBJECTS REGISTERED: 8	APPEARED: 8	PASSED: 7 TOTAL:	191	313	504		22

Aggregate (In Words) **** Five Zero Four ****

F:FAIL

124

AB:ABSENT

125

4

ID q4Enb9zegy

Quality Assurance

Quality assurance measures were implemented to uphold the accuracy and consistency of the annotated data:

- Validation Review: Annotated images underwent review by multiple team members to validate the correctness of the annotations and ensure alignment with project requirements.
- Consistency Check: Consistency checks were performed to maintain uniformity in annotation styles and labeling conventions across all images.

Export of Annotations

Upon completion of the manual annotation process, the refined annotations were exported from Label Studio:

- Export Procedure: The annotations were exported from Label Studio and saved in JSON file format for further processing and integration into the project pipeline.

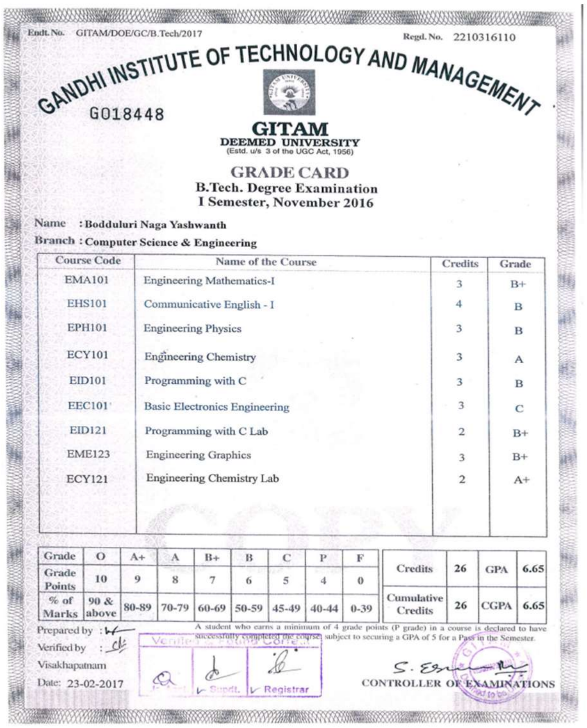
Model Training and Inference

Paddle OCR is a powerful and versatile optical character recognition (OCR) framework developed by PaddlePaddle, a deep learning platform created by Baidu. It offers a series of high-quality pre-trained models that provide an end-to-end text recognition pipeline, supporting multiple languages and a wide range of use cases, from extracting text from images and documents to recognizing handwritten and curved text.

Key features and capabilities of Paddle OCR include multilingual support, ultra-lightweight models, continuous integration and improvement, versatile use cases, and an open-source and community-driven approach. Paddle OCR supports a variety of languages, including English, Chinese, German, French, Japanese, and Korean, with more language models under development. The project can be applied to a wide range of use cases, including extracting text from forms, bills, handwritten documents, signboards, trading cards, and more.

1.Paddle OCR Inference

- Paddle OCR is a lightweight model specifically designed for extracting text from both images and PDF files.
- This model efficiently executes two primary tasks for text extraction from transcripts:
    - Text Detection: Identifying text within images, it outlines the detected text with a bounding box.
    - Text Recognition: Recognizing the text enclosed within the bounding box.
- Inference Output: Upon completion of the inference process, the model generates a text file as output. Each line in the text file corresponds to a single detection. For each detection, the output includes:
    - A 2-dimensional list containing four coordinates of the bounding box, arranged clockwise.
    - The recognized text within the bounding box.
    - The confidence score associated with the detection.



Endt. No. GITAM/DOE/GC/B.Tech/2017 Regd. No. 2210316110

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT

G018448

GITAM
DEEMED UNIVERSITY
(Estd. u/s 3 of the UGC Act, 1956)

GRADE CARD
B.Tech. Degree Examination
I Semester, November 2016

Name : Bodduluri Naga Yashwanth
Branch : Computer Science & Engineering

Course Code	Name of the Course	Credits	Grade
EMA101	Engineering Mathematics-I	3	B+
EHS101	Communicative English - I	4	B
EPH101	Engineering Physics	3	B
ECY101	Engineering Chemistry	3	A
EID101	Programming with C	3	B
EEC101	Basic Electronics Engineering	3	C
EID121	Programming with C Lab	2	B+
EME123	Engineering Graphics	3	B+
ECY121	Engineering Chemistry Lab	2	A+

Grade	O	A+	A	B+	B	C	P	F
Grade Points	10	9	8	7	6	5	4	0
% of Marks	90 & above	80-89	70-79	60-69	50-59	45-49	40-44	0-39

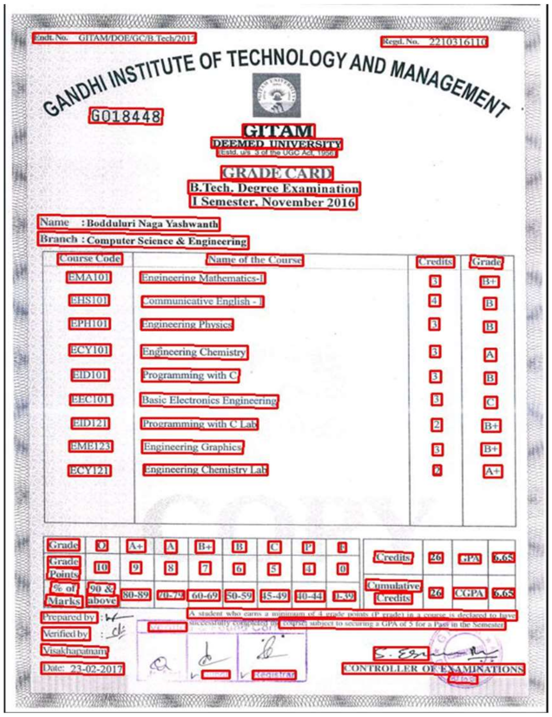
Credits	26	GPA	6.65
Cumulative Credits	26	CGPA	6.65

A student who earns a minimum of 4 grade points (P grade) in a course is declared to have successfully completed the course subject to securing a GPA of 5 for a Pass in the Semester.

Prepared by :
Verified by :
Visakhapatnam
Date: 23-02-2017

Verified & Found Correct
Supdt. Registrar

S. E
CONTROLLER OF EXAMINATIONS



Endt. No. GITAM/DOE/GC/B.Tech/2017 Regd. No. 2210316110

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT

G018448

GITAM
DEEMED UNIVERSITY
(Estd. u/s 3 of the UGC Act, 1956)

GRADE CARD
B.Tech. Degree Examination
I Semester, November 2016

Name : Bodduluri Naga Yashwanth
Branch : Computer Science & Engineering

Course Code | Name of the Course | Credits | Grade
EMA101 | Engineering Mathematics-I | 3 | B+
EHS101 | Communicative English - I | 4 | B
EPH101 | Engineering Physics | 3 | B
ECY101 | Engineering Chemistry | 3 | A
EID101 | Programming with C | 3 | B
EEC101 | Basic Electronics Engineering | 3 | C
EID121 | Programming with C Lab | 2 | B+
EME123 | Engineering Graphics | 3 | B+
ECY121 | Engineering Chemistry Lab | 2 | A+

Grade | O | A+ | A | B+ | B | C | P | F
Grade Points | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 0
% of Marks | 90 & above | 80-89 | 70-79 | 60-69 | 50-59 | 45-49 | 40-44 | 0-39

Credits 26 GPA 6.65
Cumulative Credits 26 CGPA 6.65

A student who earns a minimum of 4 grade points (P grade) in a course is declared to have successfully completed the course subject to securing a GPA of 5 for a Pass in the Semester.

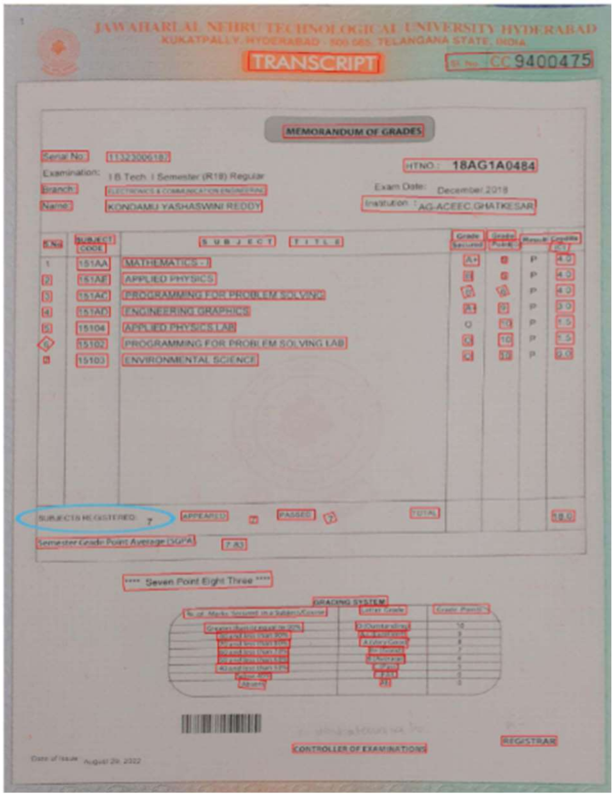
Prepared by :
Verified by :
Visakhapatnam
Date: 23-02-2017

Verified & Found Correct
Supdt. Registrar

CONTROLLER OF EXAMINATIONS

```
[[[138.0, 106.0], [461.0, 109.0], [461.0, 130.0], [138.0, 127.0]], ('Endt.No.GITAM/DOE/GC/B.Tech/2017',
0.9823161363601685)]
[[[820.0, 114.0], [1028.0, 116.0], [1028.0, 139.0], [820.0, 137.0]], ('Regd.No. 2210316110', 0.9702948331832886)]
[[[254.0, 269.0], [391.0, 269.0], [391.0, 299.0], [254.0, 299.0]], ('GO18448', 0.9655500054359436)]
[[[550.0, 300.0], [688.0, 300.0], [688.0, 330.0], [550.0, 330.0]], ('GITAM', 0.9480832815170288)]
[[[489.0, 330.0], [745.0, 333.0], [745.0, 355.0], [489.0, 351.0]], ('DEEMED UNIVERSITY', 0.9743924140930176)]
[[[499.0, 348.0], [739.0, 352.0], [738.0, 373.0], [499.0, 370.0]], ('Estd. u/s 3 of the UGC Act 1956',
0.9408138394355774)]
[[[509.0, 389.0], [727.0, 391.0], [727.0, 419.0], [509.0, 417.0]], ('GRADE CARD', 0.9641650319099426)]
[[[446.0, 421.0], [778.0, 424.0], [778.0, 452.0], [446.0, 449.0]], ('B.Tech.Degree Examination',
0.9946680665016174)]
[[[449.0, 452.0], [770.0, 455.0], [770.0, 479.0], [449.0, 475.0]], ('I Semester, November 2016',
0.9931484460830688)]
[[[152.0, 493.0], [506.0, 498.0], [505.0, 526.0], [151.0, 521.0]], ('Name: Bodduluri Naga Yashwanth',
0.969482958316803)]
[[[152.0, 529.0], [562.0, 538.0], [562.0, 566.0], [151.0, 558.0]], ('Branch:Computer Science & Engineering',
0.9796109795570374)]
[[[188.0, 571.0], [316.0, 574.0], [316.0, 598.0], [188.0, 594.0]], ('Course Code', 0.9585050940513611)]
[[[481.0, 574.0], [669.0, 578.0], [668.0, 601.0], [480.0, 597.0]], ('Name of the Course', 0.9589247107505798)]
[[[890.0, 581.0], [963.0, 581.0], [963.0, 604.0], [890.0, 604.0]], ('Credits', 0.9970329403877258)]
[[[994.0, 581.0], [1058.0, 581.0], [1058.0, 604.0], [994.0, 604.0]], ('Grade', 0.9978131055831909)]
[[[206.0, 612.0], [293.0, 612.0], [293.0, 635.0], [206.0, 635.0]], ('EMA101', 0.994859516620636)]
[[[351.0, 615.0], [592.0, 615.0], [592.0, 637.0], [351.0, 637.0]], ('Engineering Mathematics-I',
0.9909005761146545)]
```

Apart from this Enhanced output can be achieved by adjusting the confidence threshold, leading to improved results as shown below



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
KUKATPALLY, HYDERABAD - 500 085, TELANGANA STATE, INDIA

TRANSCRIPT

Sl. No. CC 9400475

MEMORANDUM OF GRADES

Serial No: 11323006187

HTNO: 18AG1A0484

Examination: I B.Tech. I Semester (R18) Regular

Exam Date: December 2018

Branch: ELECTRONICS & COMMUNICATION ENGINEERING

Name: KONDAMU YASHASWINI REDDY

Institution: AG-ACEEC,GHATKESAR

S.No	SUBJECT CODE	SUBJECT TITLE	Grade Secured	Grade Points	Result	Credits (C)
1	151AA	MATHEMATICS - I	A+	[illegible]	P	4.0
2	151AF	APPLIED PHYSICS	[illegible]	[illegible]	P	4.0
3	151AC	PROGRAMMING FOR PROBLEM SOLVING	[illegible]	[illegible]	P	4.0
4	151AD	ENGINEERING GRAPHICS	A+	9	P	3.0
5	15104	APPLIED PHYSICS LAB	O	10	P	1.5
6	15102	PROGRAMMING FOR PROBLEM SOLVING LAB	O	10	P	1.5
7	15103	ENVIRONMENTAL SCIENCE	O	10	P	0.0

SUBJECTS REGISTERED 7 APPEARED 7 PASSED 7 TOTAL 18.0

Semester Grade Point Average (SGPA) 7.83

**** Seven Point Eight Three ****

GRADING SYSTEM

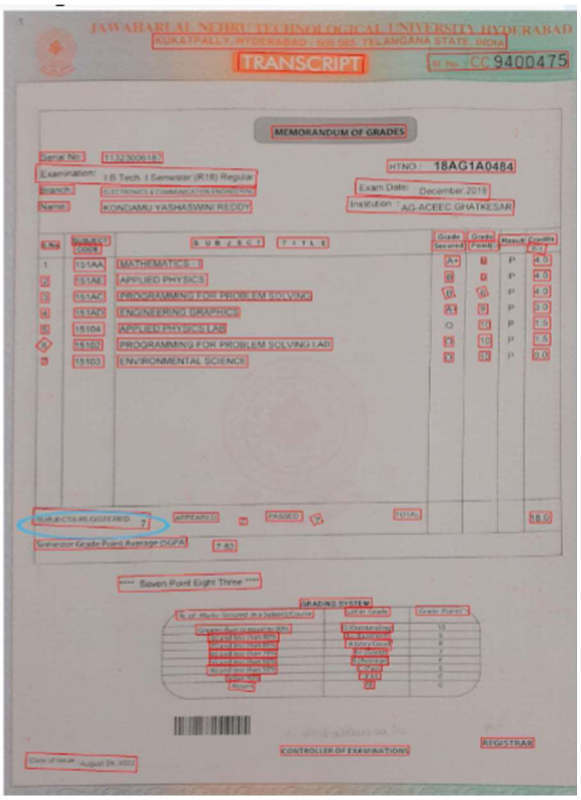
% of Marks Secured in a Subject/Course	Letter Grade	Grade Points
Greater than or equal to 90%	O (Outstanding)	10
[illegible]	[illegible]	9
[illegible]	A (Very Good)	8
[illegible]	[illegible]	7
[illegible]	[illegible]	6
[illegible]	[illegible]	5
Below 40%	F	0
Absent	Ab	0

Date of Issue August 29, 2022

CONTROLLER OF EXAMINATIONS

REGISTRAR

Model output with 0.6 confidence threshold



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
KUKATPALLY, HYDERABAD - 500 085, TELANGANA STATE, INDIA

TRANSCRIPT

Sl. No. CC 9400475

MEMORANDUM OF GRADES

Serial No: 11323006187

HTNO: 18AG1A0484

Examination: I B.Tech. I Semester (R18) Regular

Exam Date: December 2018

Branch: ELECTRONICS & COMMUNICATION ENGINEERING

Name: KONDAMU YASHASWINI REDDY

Institution: AG-ACEEC,GHATKESAR

S.No	SUBJECT CODE	SUBJECT TITLE	Grade Secured	Grade Points	Result	Credits (C)
1	151AA	MATHEMATICS - I	A+	[illegible]	P	4.0
2	151AE	APPLIED PHYSICS	[illegible]	[illegible]	P	4.0
3	151AC	PROGRAMMING FOR PROBLEM SOLVING	[illegible]	[illegible]	P	4.0
4	151AD	ENGINEERING GRAPHICS	A+	9	P	3.0
5	15104	APPLIED PHYSICS LAB	O	10	P	1.5
6	15102	PROGRAMMING FOR PROBLEM SOLVING LAB	O	10	P	1.5
7	15103	ENVIRONMENTAL SCIENCE	O	10	P	0.0

SUBJECTS REGISTERED 7 APPEARED 7 PASSED 7 TOTAL 18.0

Semester Grade Point Average (SGPA) 7.83

**** Seven Point Eight Three ****

GRADING SYSTEM

Date of Issue August 29, 2022

CONTROLLER OF EXAMINATIONS

REGISTRAR

Model output with 0.3 confidence threshold

We can observe that while Paddle OCR performs well in both detection and recognition tasks, it occasionally misses certain detections. To address this, further training of the Paddle OCR model is necessary for enhancement

2. PaddleOCR Training

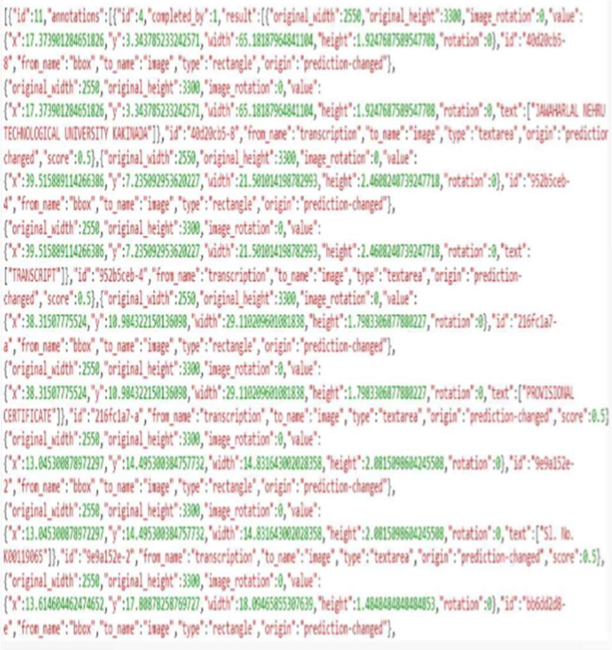
PaddleOCR offers a series of high-quality pre-trained models that provide an end-to-end text recognition pipeline. However, the framework also allows users to train custom models based on their specific requirements.

The training process for PaddleOCR models involves several key steps. First, we need to prepare their dataset, which can include images of text from Transcripts. The dataset should be annotated with the ground truth text for each image. Once the dataset is ready, we can leverage the provided training scripts and configuration files to fine-tune the pre-trained models or train new models from scratch. The training process is designed to be modular, allowing us to customize the model architecture, loss functions, and other hyperparameters to achieve the desired performance on their specific use cases

Data Preparation

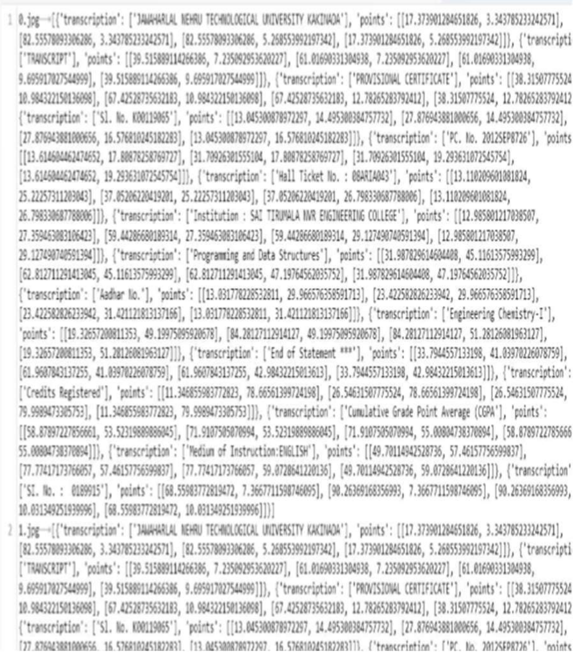
The first step in training a custom PaddleOCR model is to prepare the dataset. PaddleOCR requires the training data to be in a specific format, which includes both the image files and an annotation file.

The image files should contain the text-bearing content that the model will be trained to recognize. These can come from various sources, such as documents, signboards, or handwritten materials.
The annotation file is a text file where each line represents the annotations for a single image. Each line starts with the image filename, followed by a list of dictionaries. Each dictionary contains the bounding box coordinates and the corresponding text content for a single text instance within that image.



```
[{"id":11,"annotations":[{"id":4,"completed_by":1,"result":[{"original_width":2550,"original_height":3300,"image_rotation":0,"value":
{"x":17.373901284651826,"y":3.343785233242571,"width":65.18187964841104,"height":1.924768758954770B,"rotation":0},"id":"40d20cb5-
8","from_name":"bbox","to_name":"image","type":"rectangle","origin":"prediction-changed"},
{"original_width":2550,"original_height":3300,"image_rotation":0,"value":
{"x":17.373901284651826,"y":3.343785233242571,"width":65.18187964841104,"height":1.924768758954770B,"rotation":0,"text":["JAWAHARLAL NEHRU
TECHNOLOGICAL UNIVERSITY KAKINADA"]},"id":"40d20cb5-8","from_name":"transcription","to_name":"image","type":"textarea","origin":"predictior
changed","score":0.5},{"original_width":2550,"original_height":3300,"image_rotation":0,"value":
{"x":39.515889114266386,"y":7.235092953620227,"width":21.501014198782993,"height":2.4608240739247718,"rotation":0},"id":"952b5ceb-
4","from_name":"bbox","to_name":"image","type":"rectangle","origin":"prediction-changed"},
{"original_width":2550,"original_height":3300,"image_rotation":0,"value":
{"x":39.515889114266386,"y":7.235092953620227,"width":21.501014198782993,"height":2.4608240739247718,"rotation":0,"text":
["TRANSCRIPT"]},"id":"952b5ceb-4","from_name":"transcription","to_name":"image","type":"textarea","origin":"prediction-
changed","score":0.5},{"original_width":2550,"original_height":3300,"image_rotation":0,"value":
{"x":38.31507775524,"y":10.984322150136098,"width":29.110209601081838,"height":1.7983306877880227,"rotation":0},"id":"216fc1a7-
a","from_name":"bbox","to_name":"image","type":"rectangle","origin":"prediction-changed"},
{"original_width":2550,"original_height":3300,"image_rotation":0,"value":
{"x":38.31507775524,"y":10.984322150136098,"width":29.110209601081838,"height":1.7983306877880227,"rotation":0,"text":["PROVISIONAL
CERTIFICATE"]},"id":"216fc1a7-a","from_name":"transcription","to_name":"image","type":"textarea","origin":"prediction-changed","score":0.5}
{"original_width":2550,"original_height":3300,"image_rotation":0,"value":
{"x":13.045300878972297,"y":14.495300384757732,"width":14.831643002028358,"height":2.0815098604245508,"rotation":0},"id":"9e9a152e-
2","from_name":"bbox","to_name":"image","type":"rectangle","origin":"prediction-changed"},
{"original_width":2550,"original_height":3300,"image_rotation":0,"value":
{"x":13.045300878972297,"y":14.495300384757732,"width":14.831643002028358,"height":2.0815098604245508,"rotation":0,"text":["Sl. No.
K00119065"]},"id":"9e9a152e-2","from_name":"transcription","to_name":"image","type":"textarea","origin":"prediction-changed","score":0.5},
{"original_width":2550,"original_height":3300,"image_rotation":0,"value":
{"x":13.614604462474652,"y":17.808782587697270,"width":18.094658553076390,"height":1.4848484848484853,"rotation":0},"id":"bb6dd2d8-
e","from_name":"bbox","to_name":"image","type":"rectangle","origin":"prediction-changed"},
```

Output from label studio



```
0.jpg→[{'transcription': ['JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA'], 'points': [[17.373901284651826, 3.343785233242571],
[82.55578093306286, 3.343785233242571], [82.55578093306286, 5.268553992197342], [17.373901284651826, 5.268553992197342]]}, {'transcripti
['TRANSCRIPT'], 'points': [[39.515889114266386, 7.235092953620227], [61.01690331304938, 7.235092953620227], [61.01690331304938,
9.695917027544999], [39.515889114266386, 9.695917027544999]]}, {'transcription': ['PROVISIONAL CERTIFICATE'], 'points': [[38.31507775524
10.984322150136098], [67.42528735632183, 10.984322150136098], [67.42528735632183, 12.78265283792412], [38.31507775524, 12.78265283792412
{'transcription': ['Sl. No. K00119065'], 'points': [[13.045300878972297, 14.495300384757732], [27.876943881000656, 14.495300384757732],
[27.876943881000656, 16.57681024518228З], [13.045300878972297, 16.576810245182283]]}, {'transcription': ['PC. No. 2012SEP8726'], 'points
[[13.614604462474652, 17.80878258769727], [31.70926301555104, 17.80878258769727], [31.70926301555104, 19.293631072545754],
[13.614604462474652, 19.293631072545754]]}, {'transcription': ['Hall Ticket No. : 08ARIA043'], 'points': [[13.110209601081824,
25.22257311203043], [37.0520622041920l, 25.22257311203043], [37.05206220419201, 26.798330687788006], [13.110209601081824,
26.798330687788006]]}, {'transcription': ['Institution : SAI TIRUMALA NVR ENGINEERING COLLEGE'], 'points': [[12.985801217038507,
27.359463083106423], [59.44286680189314, 27.359463083106423], [59.44286680189314, 29.127490740591394], [12.985801217038507,
29.127490740591394]]}, {'transcription': ['Programming and Data Structures'], 'points': [[31.987829614604408, 45.11613575993299],
[62.81271129141304S, 45.11613575993299], [62.812711291413045, 47.19764562035752], [31.987829614604408, 47.19764562035752]]},
{'transcription': ['Aadhar No.'], 'points': [[13.031778228532811, 29.966576358591713], [23.422582826233942, 29.966576358591713],
[23.422582826233942, 31.421121813137166], [13.031778228532811, 31.421121813137166]]}, {'transcription': ['Engineering Chemistry-I'],
'points': [[19.32657200811353, 49.19975095920678], [84.28127112914127, 49.19975095920678], [84.28127112914127, 51.28126081963127],
[19.32657200811353, 51.28126081963127]]}, {'transcription': ['End of Statement ***'], 'points': [[33.7944557133198, 41.03970226078759],
[61.9607843137255, 41.03970226078759], [61.9607843137255, 42.98432215013613], [33.7944557133198, 42.98432215013613]]}, {'transcription':
['Credits Registered'], 'points': [[11.346855983772823, 78.66561399724198], [26.5463150777S524, 78.66561399724198], [26.5463150777S524,
79.9989473305753], [11.346855983772823, 79.9989473305753]]}, {'transcription': ['Cumulative Grade Point Average (CGPA'], 'points':
[[58.8789722785666l, 53.52319889886045], [71.9107505070994, 53.52319889886045], [71.9107505070994, 55.00804738370894], [58.878972278566б
55.00804738370894]]}, {'transcription': ['Medium of Instruction:ENGLISH'], 'points': [[49.7011494252873б, 57.46157756599837],
[77.77417173766057, 57.46157756599837], [77.77417173766057, 59.0728641220136], [49.70114942528736, 59.0728641220136]]}, {'transcription'
['SI. No. :  0189915'], 'points': [[68.55983772819472, 7.366771159874б095], [90.26369168356993, 7.3667711598746095], [90.26369168356993,
10.031349251939996], [68.55983772819472, 10.031349251939996]]}]
1.jpg→[{'transcription': ['JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA'], 'points': [[17.373901284651826, 3.343785233242571],
[82.55578093306286, 3.343785233242571], [82.55578093306286, 5.268553992197342], [17.373901284651826, 5.268553992197342]]}, {'transcripti
['TRANSCRIPT'], 'points': [[39.515889114266386, 7.235092953620227], [61.01690331304938, 7.235092953620227], [61.01690331304938,
9.695917027544999], [39.515889114266386, 9.695917027544999]]}, {'transcription': ['PROVISIONAL CERTIFICATE'], 'points': [[38.31507775524
10.984322150136098], [67.42528735632183, 10.984322150136098], [67.42528735632183, 12.78265283792412], [38.31507775524, 12.78265283792412
{'transcription': ['Sl. No. K00119065'], 'points': [[13.045300878972297, 14.495300384757732], [27.876943881000656, 14.495300384757732],
[27.876943881000656, 16.576810245182283], [13.045300878972297, 16.576810245182283]]}, {'transcription': ['PC. No. 2012SEP8726'], 'points
```

Input to PaddleOCR

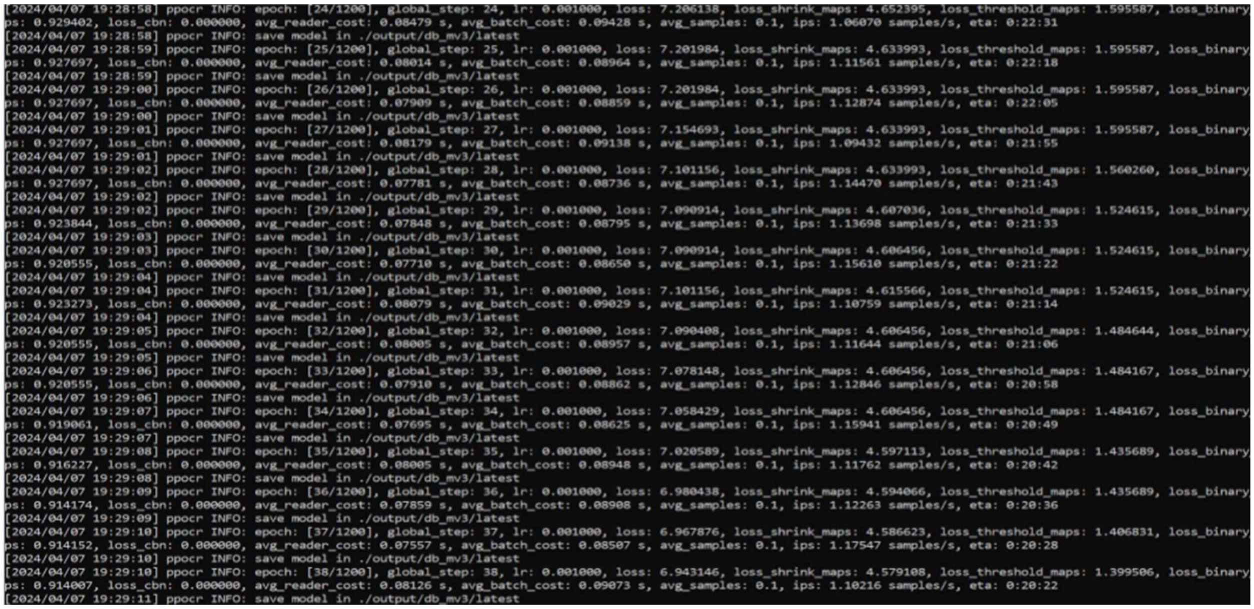
Model Fine Tuning

The Paddle OCR model that is fine-tuned on the transcript dataset is called PPOCRv3. This model is built upon the PaddleOCR framework and incorporates several architectural improvements over previous versions.
PPOCRv3 utilizes a multi-stage text detection and recognition pipeline. The text detection stage employs a lightweight yet powerful detection model to identify the locations of text within the input images. The text recognition stage then applies an advanced recognition model to accurately decode the text content within the detected regions.

The PPOCRv3 architecture also includes various optimization techniques, such as model compression and quantization, to ensure the model remains lightweight and efficient, making it suitable for deployment on a wide range of devices, from high-end servers to edge devices.

Model Training details: The PPOCRv3 model was trained on the transcript dataset with the following hyperparameters:

- Training epochs: 1200
- Batch size: 64
- Learning rate: 0.001
- Dataset size: 150 samples
- Pre-trained model: PPOCRv3



```
[2024/04/07 19:28:58] ppocr INFO: epoch: [24/1200], global_step: 24, lr: 0.001000, loss: 7.206138, loss_shrink_maps: 4.652395, loss_threshold_maps: 1.595587, loss_binary
ps: 0.929402, loss_cbn: 0.000000, avg_reader_cost: 0.08479 s, avg_batch_cost: 0.09428 s, avg_samples: 0.1, ips: 1.06070 samples/s, eta: 0:22:31
[2024/04/07 19:28:58] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:28:59] ppocr INFO: epoch: [25/1200], global_step: 25, lr: 0.001000, loss: 7.201984, loss_shrink_maps: 4.633993, loss_threshold_maps: 1.595587, loss_binary
ps: 0.927697, loss_cbn: 0.000000, avg_reader_cost: 0.08014 s, avg_batch_cost: 0.08964 s, avg_samples: 0.1, ips: 1.11561 samples/s, eta: 0:22:18
[2024/04/07 19:28:59] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:00] ppocr INFO: epoch: [26/1200], global_step: 26, lr: 0.001000, loss: 7.201984, loss_shrink_maps: 4.633993, loss_threshold_maps: 1.595587, loss_binary
ps: 0.927697, loss_cbn: 0.000000, avg_reader_cost: 0.07909 s, avg_batch_cost: 0.08859 s, avg_samples: 0.1, ips: 1.12874 samples/s, eta: 0:22:05
[2024/04/07 19:29:00] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:01] ppocr INFO: epoch: [27/1200], global_step: 27, lr: 0.001000, loss: 7.154693, loss_shrink_maps: 4.633993, loss_threshold_maps: 1.595587, loss_binary
ps: 0.927697, loss_cbn: 0.000000, avg_reader_cost: 0.08179 s, avg_batch_cost: 0.09138 s, avg_samples: 0.1, ips: 1.09432 samples/s, eta: 0:21:55
[2024/04/07 19:29:01] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:02] ppocr INFO: epoch: [28/1200], global_step: 28, lr: 0.001000, loss: 7.101156, loss_shrink_maps: 4.633993, loss_threshold_maps: 1.560260, loss_binary
ps: 0.927697, loss_cbn: 0.000000, avg_reader_cost: 0.07781 s, avg_batch_cost: 0.08736 s, avg_samples: 0.1, ips: 1.14470 samples/s, eta: 0:21:43
[2024/04/07 19:29:02] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:02] ppocr INFO: epoch: [29/1200], global_step: 29, lr: 0.001000, loss: 7.090914, loss_shrink_maps: 4.607036, loss_threshold_maps: 1.524615, loss_binary
ps: 0.923844, loss_cbn: 0.000000, avg_reader_cost: 0.07848 s, avg_batch_cost: 0.08795 s, avg_samples: 0.1, ips: 1.13698 samples/s, eta: 0:21:33
[2024/04/07 19:29:03] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:03] ppocr INFO: epoch: [30/1200], global_step: 30, lr: 0.001000, loss: 7.090914, loss_shrink_maps: 4.606456, loss_threshold_maps: 1.524615, loss_binary
ps: 0.920555, loss_cbn: 0.000000, avg_reader_cost: 0.07710 s, avg_batch_cost: 0.08650 s, avg_samples: 0.1, ips: 1.15610 samples/s, eta: 0:21:22
[2024/04/07 19:29:04] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:04] ppocr INFO: epoch: [31/1200], global_step: 31, lr: 0.001000, loss: 7.101156, loss_shrink_maps: 4.615566, loss_threshold_maps: 1.524615, loss_binary
ps: 0.923273, loss_cbn: 0.000000, avg_reader_cost: 0.08079 s, avg_batch_cost: 0.09029 s, avg_samples: 0.1, ips: 1.10759 samples/s, eta: 0:21:14
[2024/04/07 19:29:04] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:05] ppocr INFO: epoch: [32/1200], global_step: 32, lr: 0.001000, loss: 7.090408, loss_shrink_maps: 4.606456, loss_threshold_maps: 1.484644, loss_binary
ps: 0.920555, loss_cbn: 0.000000, avg_reader_cost: 0.08005 s, avg_batch_cost: 0.08957 s, avg_samples: 0.1, ips: 1.11644 samples/s, eta: 0:21:06
[2024/04/07 19:29:05] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:06] ppocr INFO: epoch: [33/1200], global_step: 33, lr: 0.001000, loss: 7.078148, loss_shrink_maps: 4.606456, loss_threshold_maps: 1.484167, loss_binary
ps: 0.920555, loss_cbn: 0.000000, avg_reader_cost: 0.07910 s, avg_batch_cost: 0.08862 s, avg_samples: 0.1, ips: 1.12846 samples/s, eta: 0:20:58
[2024/04/07 19:29:06] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:07] ppocr INFO: epoch: [34/1200], global_step: 34, lr: 0.001000, loss: 7.058429, loss_shrink_maps: 4.606456, loss_threshold_maps: 1.484167, loss_binary
ps: 0.919061, loss_cbn: 0.000000, avg_reader_cost: 0.07695 s, avg_batch_cost: 0.08625 s, avg_samples: 0.1, ips: 1.15941 samples/s, eta: 0:20:49
[2024/04/07 19:29:07] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:08] ppocr INFO: epoch: [35/1200], global_step: 35, lr: 0.001000, loss: 7.020589, loss_shrink_maps: 4.597113, loss_threshold_maps: 1.435689, loss_binary
ps: 0.916227, loss_cbn: 0.000000, avg_reader_cost: 0.08005 s, avg_batch_cost: 0.08948 s, avg_samples: 0.1, ips: 1.11762 samples/s, eta: 0:20:42
[2024/04/07 19:29:08] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:09] ppocr INFO: epoch: [36/1200], global_step: 36, lr: 0.001000, loss: 6.980438, loss_shrink_maps: 4.594066, loss_threshold_maps: 1.435689, loss_binary
ps: 0.914174, loss_cbn: 0.000000, avg_reader_cost: 0.07859 s, avg_batch_cost: 0.08908 s, avg_samples: 0.1, ips: 1.12263 samples/s, eta: 0:20:36
[2024/04/07 19:29:09] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:10] ppocr INFO: epoch: [37/1200], global_step: 37, lr: 0.001000, loss: 6.967876, loss_shrink_maps: 4.586623, loss_threshold_maps: 1.406831, loss_binary
ps: 0.914152, loss_cbn: 0.000000, avg_reader_cost: 0.07557 s, avg_batch_cost: 0.08507 s, avg_samples: 0.1, ips: 1.17547 samples/s, eta: 0:20:28
[2024/04/07 19:29:10] ppocr INFO: save model in ./output/db_mv3/latest
[2024/04/07 19:29:10] ppocr INFO: epoch: [38/1200], global_step: 38, lr: 0.001000, loss: 6.943146, loss_shrink_maps: 4.579108, loss_threshold_maps: 1.399506, loss_binary
ps: 0.914007, loss_cbn: 0.000000, avg_reader_cost: 0.08126 s, avg_batch_cost: 0.09073 s, avg_samples: 0.1, ips: 1.10216 samples/s, eta: 0:20:22
[2024/04/07 19:29:11] ppocr INFO: save model in ./output/db_mv3/latest
```

Text extraction and recognition results have significantly improved as a result of the Paddle OCR model's fine-tuning on the transcript dataset. The optimized PPOCRv3 model performs better in terms of reliably identifying and classifying text within transcript-related content because of its increased robustness and efficiency, as well as its capacity to learn common patterns and structures found in transcripts. The quality and accuracy of the text processing capabilities have generally improved as a result of the training process, which makes the Paddle OCR model a more practical choice for tasks involving transcripts.

Model Evaluation

Mean IoU (Intersection over Union)

Intersection over Union (IoU) is a widely used metric for evaluating object detection and semantic segmentation models. It measures the overlap between the predicted bounding box or segmentation mask and the ground truth. The mean IoU (mIoU) is the average IoU across all classes, providing an overall measure of the model's segmentation accuracy. A higher mIoU indicates better localization of the predicted objects or segments.

Classification Accuracy

In addition to mIoU, classification accuracy is another important metric for evaluating semantic segmentation models. It measures the percentage of pixels that are correctly classified by the model, regardless of their location. Classification accuracy provides a complementary perspective to mIoU, as it focuses on the model's ability to assign the correct class label to each pixel.

Together, mIoU and classification accuracy give a comprehensive view of the model's performance. Analyzing these metrics can help identify the model's strengths and weaknesses, guiding further improvements to the model's design and training.

According to the output, we observed the following metrics

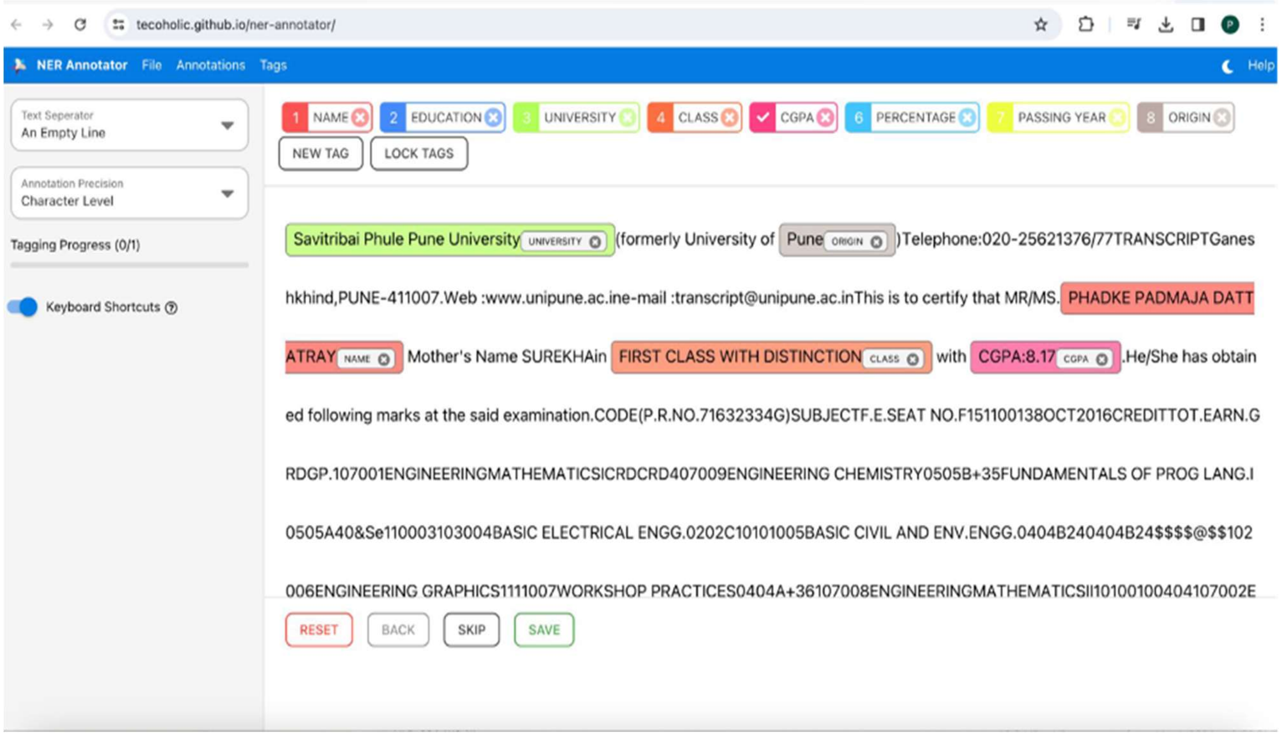
PaddleOCR	Classification Accuracy	Mean IOU
Original Model	65.73	80.34
Fine Tuned Model	66.67	83.57
Fine Tuned Model(Only Grades)	83.3	48.9

The performance metrics provided for the PaddleOCR model show some interesting insights. The original model had a classification accuracy of 65.73% and a mean IoU of 80.34%, indicating reasonably good performance in both correctly classifying pixels and accurately localizing the text regions. When the model was fine-tuned, the classification accuracy improved to 66.67% and the mean IoU increased to 83.57%, suggesting that the fine-tuning process enhanced the model's overall text extraction and recognition capabilities. However, when the fine-tuned model was evaluated on only the "grades" subset of the data, the classification accuracy increased significantly to 83.3%, but the mean IoU dropped to 48.9%. This indicates that while the model became more adept at correctly identifying the class labels for grade-related content, its ability to accurately localize the text regions in those cases decreased

Conversion of Text to JSON file:

1.NER and spaCy Model:

- We began by converting the PDF file into images and utilized Paddle OCR to extract text from these images.
- Named Entity Recognition (NER) was employed, a natural language processing (NLP) technique aimed at identifying and categorizing named entities in text.
- NER's primary objective is to extract structured information from unstructured text by categorizing entities into predefined groups.Both NER and the Spacy model were experimented with to extract key values from the obtained text.
- Labeled data was curated using an NER annotator tool, resulting in the generation of a JSON file. Utilizing the spaCy library, we applied NER on the extracted text to identify relevant information.
- While the NER model successfully retrieved values for categories such as Name, University Name, and CGPA or percentages, it encountered limitations in extracting details for Full Subjects and their respective grades.
- This method proved ineffective in capturing these specific details.



Text Seperator: An Empty Line

Annotation Precision: Character Level

Savitribai Phule Pune University UNIVERSITY (formerly University of Pune ORIGIN)Telephone:020-25621376/77TRANSCRIPTGanes

hkhind,PUNE-411007.Web :www.unipune.ac.ine-mail :transcript@unipune.ac.inThis is to certify that MR/MS. PHADKE PADMAJA DATT

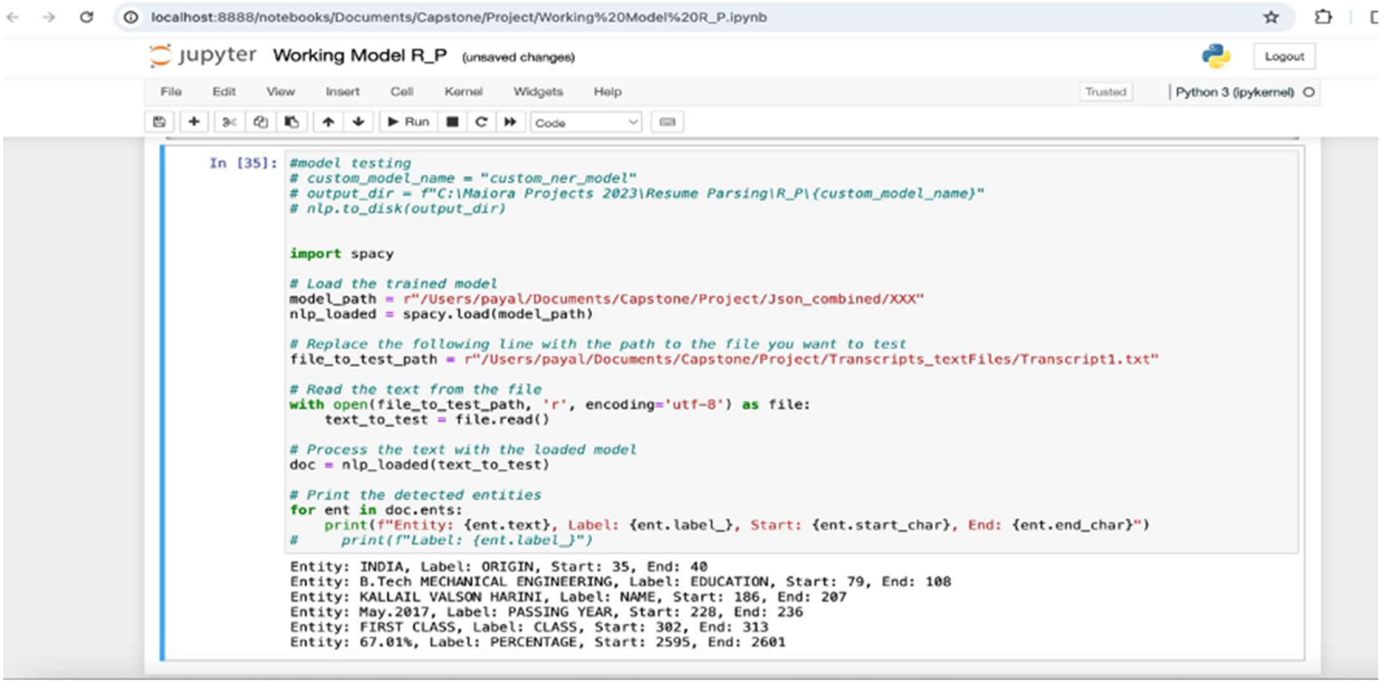
ATRAY NAME Mother's Name SUREKHAin FIRST CLASS WITH DISTINCTION CLASS with CGPA:8.17 CGPA .He/She has obtain

ed following marks at the said examination.CODE(P.R.NO.71632334G)SUBJECTF.E.SEAT NO.F151100138OCT2016CREDITTOT.EARN.G

RDGP.107001ENGINEERINGMATHEMATICSICRDCRD407009ENGINEERING CHEMISTRY0505B+35FUNDAMENTALS OF PROG LANG.I

0505A40&Se110003103004BASIC ELECTRICAL ENGG.0202C10101005BASIC CIVIL AND ENV.ENGG.0404B240404B24$$$$@$$102

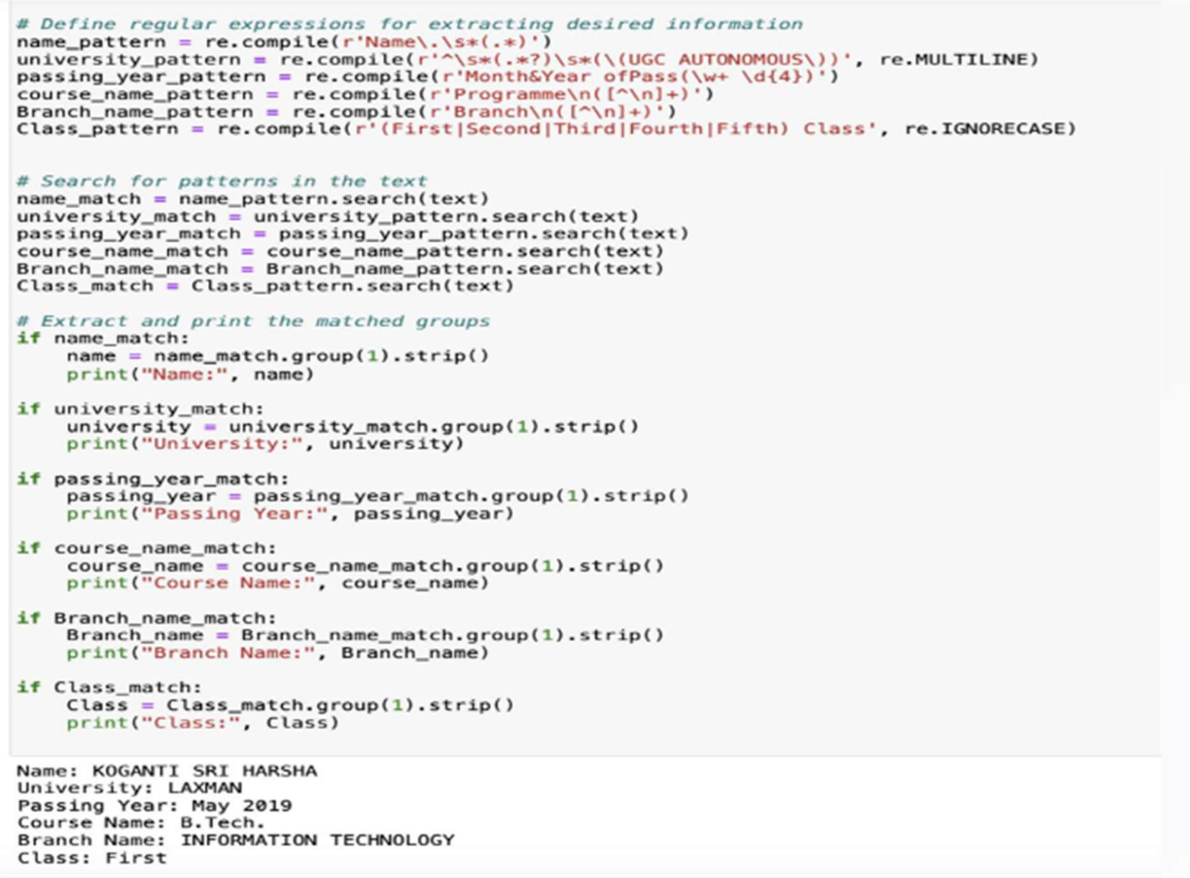
006ENGINEERING GRAPHICS1111007WORKSHOP PRACTICES0404A+36107008ENGINEERINGMATHEMATICSII10100100404107002E


localhost:8888/notebooks/Documents/Capstone/Project/Working%20Model%20R_P.ipynb
jupyter
Working Model R_P (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [35]: #model testing
         # custom_model_name = "custom_ner_model"
         # output_dir = f"C:\Maiora Projects 2023\Resume Parsing\R_P\{custom_model_name}"
         # nlp.to_disk(output_dir)


         import spacy

         # Load the trained model
         model_path = r"/Users/payal/Documents/Capstone/Project/Json_combined/XXX"
         nlp_loaded = spacy.load(model_path)

         # Replace the following line with the path to the file you want to test
         file_to_test_path = r"/Users/payal/Documents/Capstone/Project/Transcripts_textFiles/Transcript1.txt"

         # Read the text from the file
         with open(file_to_test_path, 'r', encoding='utf-8') as file:
             text_to_test = file.read()

         # Process the text with the loaded model
         doc = nlp_loaded(text_to_test)

         # Print the detected entities
         for ent in doc.ents:
             print(f"Entity: {ent.text}, Label: {ent.label_}, Start: {ent.start_char}, End: {ent.end_char}")
         #     print(f"Label: {ent.label_}")
         Entity: INDIA, Label: ORIGIN, Start: 35, End: 40
         Entity: B.Tech MECHANICAL ENGINEERING, Label: EDUCATION, Start: 79, End: 108
         Entity: KALLAIL VALSON HARINI, Label: NAME, Start: 186, End: 207
         Entity: May.2017, Label: PASSING YEAR, Start: 228, End: 236
         Entity: FIRST CLASS, Label: CLASS, Start: 302, End: 313
         Entity: 67.01%, Label: PERCENTAGE, Start: 2595, End: 2601
```

2. Regular Expression:

- Our attempt to employ regular expressions to extract meaningful data encountered challenges.
- Specifically, extracting data from tables, such as subjects and their corresponding grades, proved difficult.
- The tabular format of the data caused it to be extracted line by line, with coordinate points providing additional complexity.
- Despite our efforts to utilize regular expressions along with coordinates, we were unable to successfully extract subject and grade information using this approach.



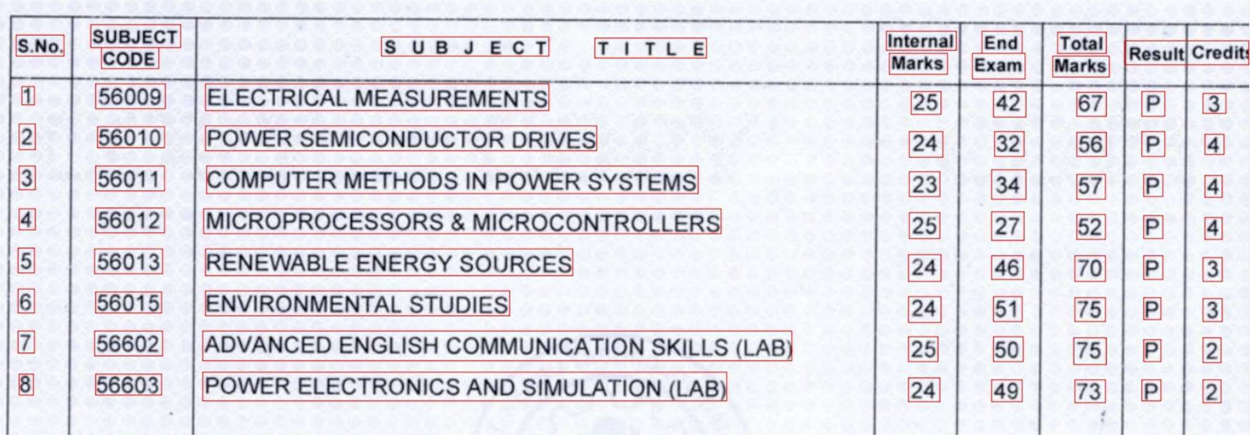
```
# Define regular expressions for extracting desired information
name_pattern = re.compile(r'Name\.\s*(.*)')
university_pattern = re.compile(r'^\s*(.*?)\s*(\(UGC AUTONOMOUS\))', re.MULTILINE)
passing_year_pattern = re.compile(r'Month&Year ofPass(\w+ \d{4})')
course_name_pattern = re.compile(r'Programme\n([^\n]+)')
Branch_name_pattern = re.compile(r'Branch\n([^\n]+)')
Class_pattern = re.compile(r'(First|Second|Third|Fourth|Fifth) Class', re.IGNORECASE)


# Search for patterns in the text
name_match = name_pattern.search(text)
university_match = university_pattern.search(text)
passing_year_match = passing_year_pattern.search(text)
course_name_match = course_name_pattern.search(text)
Branch_name_match = Branch_name_pattern.search(text)
Class_match = Class_pattern.search(text)

# Extract and print the matched groups
if name_match:
    name = name_match.group(1).strip()
    print("Name:", name)

if university_match:
    university = university_match.group(1).strip()
    print("University:", university)

if passing_year_match:
    passing_year = passing_year_match.group(1).strip()
    print("Passing Year:", passing_year)

if course_name_match:
    course_name = course_name_match.group(1).strip()
    print("Course Name:", course_name)

if Branch_name_match:
    Branch_name = Branch_name_match.group(1).strip()
    print("Branch Name:", Branch_name)

if Class_match:
    Class = Class_match.group(1).strip()
    print("Class:", Class)
```

```
Name: KOGANTI SRI HARSHA
University: LAXMAN
Passing Year: May 2019
Course Name: B.Tech.
Branch Name: INFORMATION TECHNOLOGY
Class: First
```

3. Text Detection and Recognition, Vertical and horizontal lines and used Non-Maximum Suppression Technique

Text Detection and Recognition:

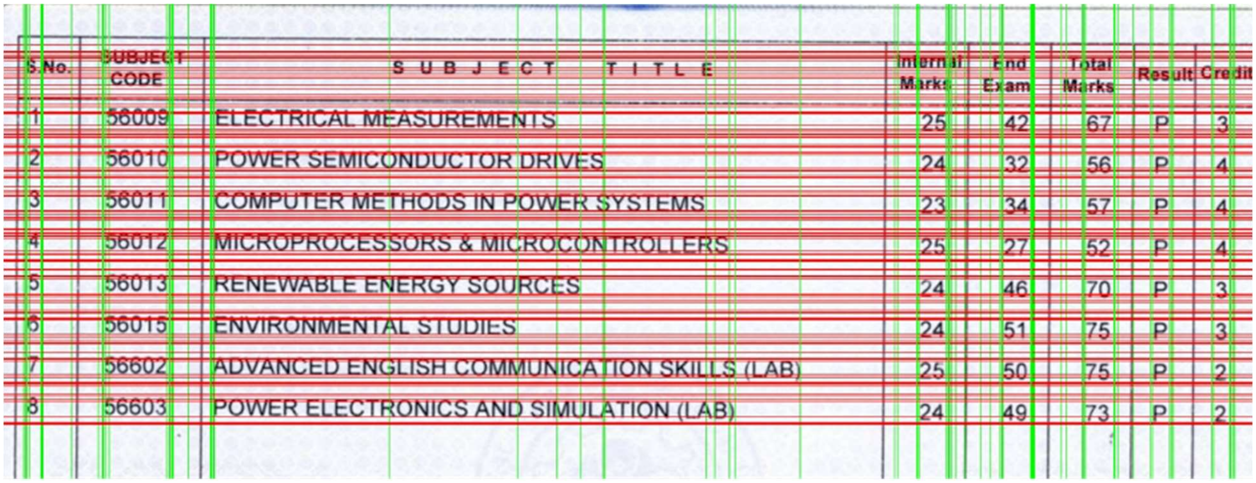
With the Paddle OCR model, we identify and recognize text.



S.No.	SUBJECT CODE	SUBJECT TITLE	Internal Marks	End Exam	Total Marks	Result	Credits
1	56009	ELECTRICAL MEASUREMENTS	25	42	67	P	3
2	56010	POWER SEMICONDUCTOR DRIVES	24	32	56	P	4
3	56011	COMPUTER METHODS IN POWER SYSTEMS	23	34	57	P	4
4	56012	MICROPROCESSORS & MICROCONTROLLERS	25	27	52	P	4
5	56013	RENEWABLE ENERGY SOURCES	24	46	70	P	3
6	56015	ENVIRONMENTAL STUDIES	24	51	75	P	3
7	56602	ADVANCED ENGLISH COMMUNICATION SKILLS (LAB)	25	50	75	P	2
8	56603	POWER ELECTRONICS AND SIMULATION (LAB)	24	49	73	P	2

To get Horizontal and Vertical Lines:

- The code processes text output containing coordinates and their corresponding threshold scores.
- It utilizes this information to create vertical and horizontal lines, effectively constructing a table within the transcript image.
- The code iterates over a list of bounding boxes (boxes).
- For each bounding box, it calculates the coordinates and dimensions of two new bounding boxes: one aligned horizontally (horiz_boxes) and one aligned vertically (vert_boxes).
- The code draws rectangles for each of these new bounding boxes on an image (im) using OpenCV (cv2.rectangle).
- Horizontal bounding boxes are drawn in red, while vertical bounding boxes are drawn in green.
- The visualization aids in understanding the alignment and dimensions of the bounding boxes relative to the image.

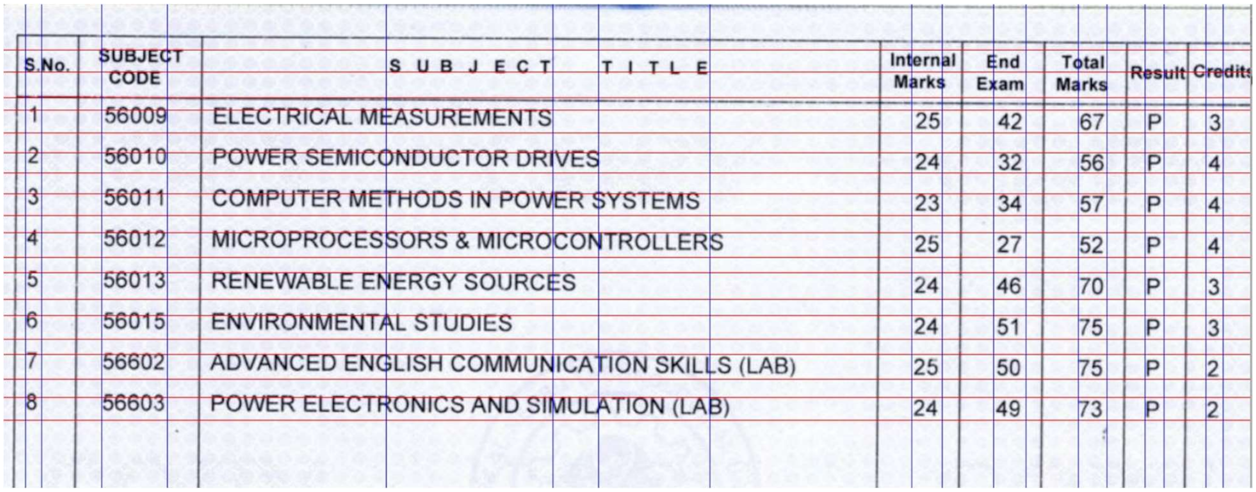


S.No.	SUBJECT CODE	SUBJECT TITLE	Internal Marks	End Exam	Total Marks	Result	Credit
1	56009	ELECTRICAL MEASUREMENTS	25	42	67	P	3
2	56010	POWER SEMICONDUCTOR DRIVES	24	32	56	P	4
3	56011	COMPUTER METHODS IN POWER SYSTEMS	23	34	57	P	4
4	56012	MICROPROCESSORS & MICROCONTROLLERS	25	27	52	P	4
5	56013	RENEWABLE ENERGY SOURCES	24	46	70	P	3
6	56015	ENVIRONMENTAL STUDIES	24	51	75	P	3
7	56602	ADVANCED ENGLISH COMMUNICATION SKILLS (LAB)	25	50	75	P	2
8	56603	POWER ELECTRONICS AND SIMULATION (LAB)	24	49	73	P	2

Non-Max Suppression:

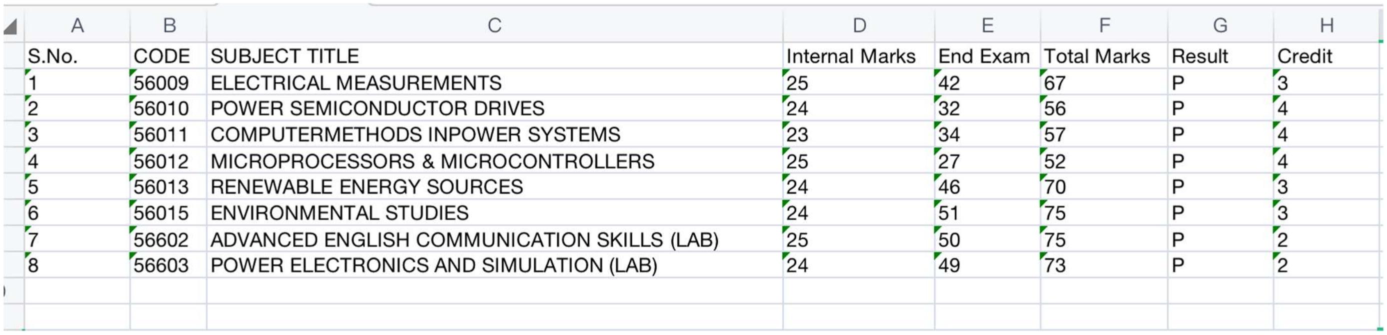
- The code utilizes TensorFlow's tf.image.non_max_suppression function to perform non-maximum suppression (NMS) on horizontal and vertical bounding boxes.
- For horizontal and vertical bounding boxes, NMS is applied with specific parameters such as maximum output size, IoU threshold, and score threshold.
- Retained bounding box indices are stored in horiz_out and vert_out variables for horizontal and vertical boxes, respectively.
- Red and blue rectangles are drawn on the image for retained horizontal and vertical bounding boxes, respectively.

- Below is the resulting image with drawn bounding boxes.



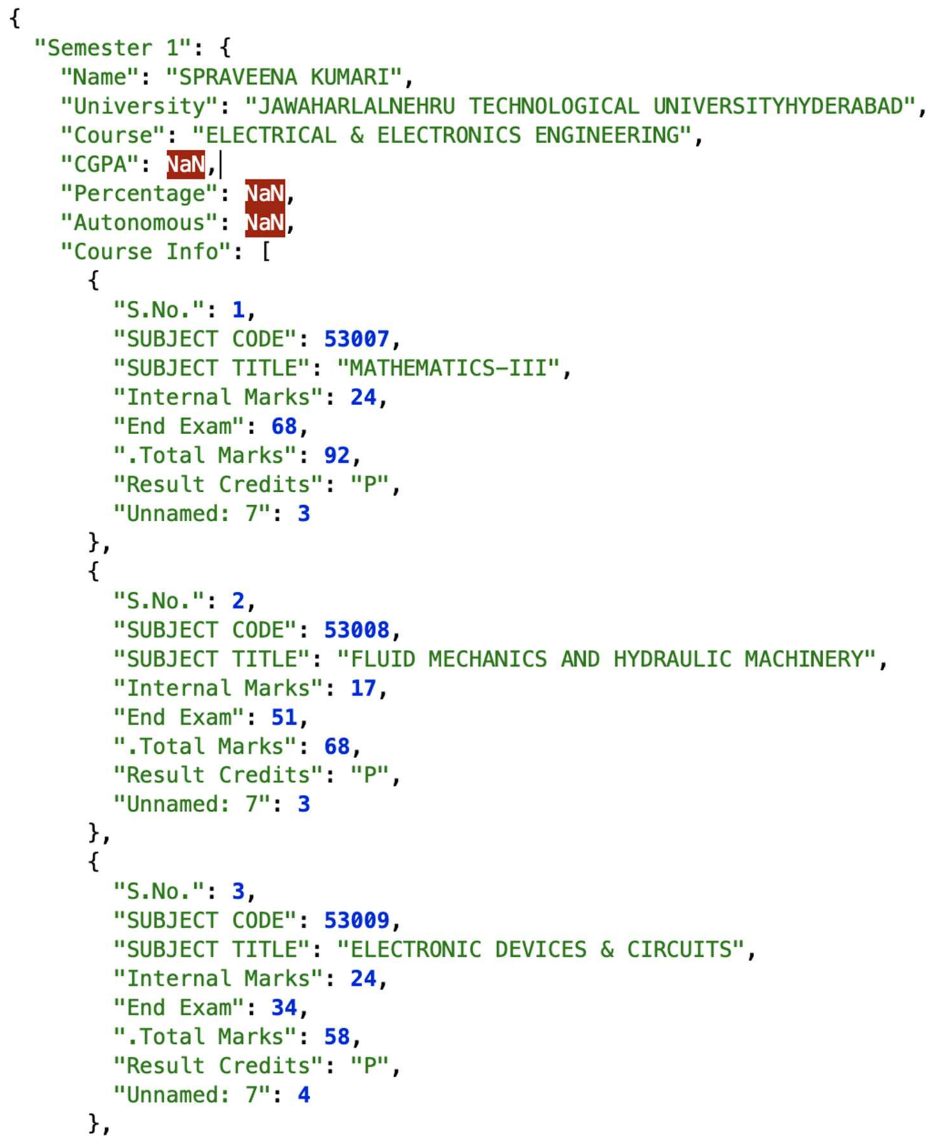
S.No.	SUBJECT CODE	S U B J E C T T I T L E	Internal Marks	End Exam	Total Marks	Result	Credits
1	56009	ELECTRICAL MEASUREMENTS	25	42	67	P	3
2	56010	POWER SEMICONDUCTOR DRIVES	24	32	56	P	4
3	56011	COMPUTER METHODS IN POWER SYSTEMS	23	34	57	P	4
4	56012	MICROPROCESSORS & MICROCONTROLLERS	25	27	52	P	4
5	56013	RENEWABLE ENERGY SOURCES	24	46	70	P	3
6	56015	ENVIRONMENTAL STUDIES	24	51	75	P	3
7	56602	ADVANCED ENGLISH COMMUNICATION SKILLS (LAB)	25	50	75	P	2
8	56603	POWER ELECTRONICS AND SIMULATION (LAB)	24	49	73	P	2

- The code initially addressed the detection of overlapping areas within blue-line columns.
- However, it did not account for checking duplicate texts inside the detection boxes.
- To address this, the data was added to an Excel sheet.



A	B	C	D	E	F	G	H
S.No.	CODE	SUBJECT TITLE	Internal Marks	End Exam	Total Marks	Result	Credit
1	56009	ELECTRICAL MEASUREMENTS	25	42	67	P	3
2	56010	POWER SEMICONDUCTOR DRIVES	24	32	56	P	4
3	56011	COMPUTERMETHODS INPOWER SYSTEMS	23	34	57	P	4
4	56012	MICROPROCESSORS & MICROCONTROLLERS	25	27	52	P	4
5	56013	RENEWABLE ENERGY SOURCES	24	46	70	P	3
6	56015	ENVIRONMENTAL STUDIES	24	51	75	P	3
7	56602	ADVANCED ENGLISH COMMUNICATION SKILLS (LAB)	25	50	75	P	2
8	56603	POWER ELECTRONICS AND SIMULATION (LAB)	24	49	73	P	2

- Subsequently, separate code was implemented to eliminate duplicate rows and columns from the Excel file.
- Finally, the table was converted to JSON format for further processing.



```
{
  "Semester 1": {
    "Name": "SPRAVEENA KUMARI",
    "University": "JAWAHARLALNEHRU TECHNOLOGICAL UNIVERSITYHYDERABAD",
    "Course": "ELECTRICAL & ELECTRONICS ENGINEERING",
    "CGPA": NaN,
    "Percentage": NaN,
    "Autonomous": NaN,
    "Course Info": [
      {
        "S.No.": 1,
        "SUBJECT CODE": 53007,
        "SUBJECT TITLE": "MATHEMATICS-III",
        "Internal Marks": 24,
        "End Exam": 68,
        ".Total Marks": 92,
        "Result Credits": "P",
        "Unnamed: 7": 3
      },
      {
        "S.No.": 2,
        "SUBJECT CODE": 53008,
        "SUBJECT TITLE": "FLUID MECHANICS AND HYDRAULIC MACHINERY",
        "Internal Marks": 17,
        "End Exam": 51,
        ".Total Marks": 68,
        "Result Credits": "P",
        "Unnamed: 7": 3
      },
      {
        "S.No.": 3,
        "SUBJECT CODE": 53009,
        "SUBJECT TITLE": "ELECTRONIC DEVICES & CIRCUITS",
        "Internal Marks": 24,
        "End Exam": 34,
        ".Total Marks": 58,
        "Result Credits": "P",
        "Unnamed: 7": 4
      },
```

3. Discussion

i) Challenges Faced:

PaddleOCR Training

- The challenge of acquiring adequate computational resources for training was resolved with the support of CITlab.
- Securing a sufficient amount of annotated data was a prerequisite for initiating training on PaddleOCR.

Text to JSON File:

- Attempted various methods like NER and regular expressions to convert text data to JSON, but faced limitations.
- Transitioned to extracting data using horizontal and vertical lines, which proved effective.

Handling Duplicates in Data (Excel File):

- Encountered challenges in removing duplicates from vertical and horizontal lines.
- Initially tried removing duplicates while checking overlapping areas in the lines.
- Difficulties arose in accurately assessing overlapping between vertical lines and their detection boxes, as well as identifying duplicate strings within these boxes across multiple horizontal lines.
- Despite efforts, this approach was unsuccessful.
- Opted to focus on methods involving overlapping intersection and intersection over union.
- Devised a different logic to identify and handle duplicate strings within boxes after they were added to the Excel sheet.

(ii) things that you think could have been done differently.

PaddleOCR Training:

- We could have trained a much more effective model with more data.
- It is also possible to better generalize the model by conducting more thorough testing and hyperparameter tuning.

Text to Excel:

- Consider incorporating logic for removing duplicates from rows and columns during the creation of horizontal and vertical columns.
- Instead of separately processing regular information like Name and University Name and performing different methods on cropped images containing only tables with subjects and their grades, integrate these processes for efficiency.